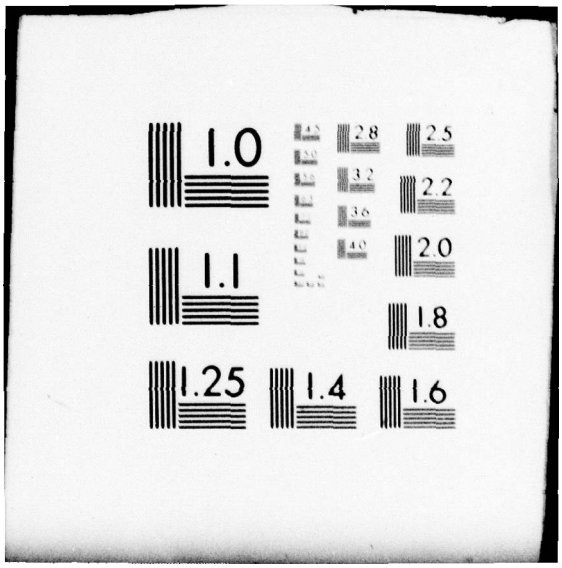
AD-A075 461 YEAGER (ROBERT F) CHAMPAIGN IL F/G 5/9
COMPUTER-BASED ELECTRONICS MAINTENANCE TRAINING: DESCRIPTION OF--ETC(U)
SEP 79 R F YEAGER , Z M SIMUTIS MDA903-78-M-AA25
UNCLASSIFIED ARI-RP-79-10 NL

| OF |
AD
A075461

END
DATE
FILMED
11-79
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

LEVEL II

Research Product 79-10

Computer-Based Electronics Maintenance Training: Description of Lessons

Educational Technology and Simulation Technical Area

September 1979

U.S. ARMY RESEARCH INSTITUTE for the BEHAVIORAL and SOCIAL SCIENCES

C-2

U. S. ARMY RESEARCH INSTITUTE
FOR THE BEHAVIORAL AND SOCIAL SCIENCES

A Field Operating Agency under the Jurisdiction of the
Deputy Chief of Staff for Personnel

JOSEPH ZEIDNER
Technical Director

WILLIAM L. HAUSER
Colonel, US Army
Commander

NOTICES

FINAL DISPOSITION: This Research Product may be destroyed when it is no longer needed. Please do not return it to the U.S. Army Research Institute for the Behavioral and Social Sciences.

NOTE: This Research Product is not to be construed as an official Department of the Army document in its present form.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: RP-79-10

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle): Computer-Based Electronics Maintenance Training: Description of Lessons

5. TYPE OF REPORT & PERIOD COVERED: Final Report, Jul 1978 — Jan 1979

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s): Robert F. Yeager, Champaign, IL
Zita M. Simutis, Army Research Institute

8. CONTRACT OR GRANT NUMBER(s): MDA 903-78-M-AA25

9. PERFORMING ORGANIZATION NAME AND ADDRESS: Robert F. Yeager (Robert F.), New
508 West John Street
Champaign, IL 61820

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 2Q162722A764

11. CONTROLLING OFFICE NAME AND ADDRESS
US Army Research Institute for the Behavioral and Social Sciences
5001 Eisenhower Ave., Alexandria, VA 22333

12. REPORT DATE: September 1979

13. NUMBER OF PAGES: 39

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): ARI

15. SECURITY CLASS. (of this report): Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer-Based Education
Game-Based Instruction
Electronics Maintenance
Electronics Training
PLATO IV
Electronic Troubleshooting

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A series of computer-based lessons were developed on the University of Illinois PLATO System for use in research in game-based training. These lessons provide instruction and practice in reading logic symbols, logic diagrams, and electronics troubleshooting. The purpose of this report is to describe the available lessons and management routines to interested instructors and trainers of electronic maintenance.

411 421

JCB

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

COMPUTER-BASED ELECTRONICS MAINTENANCE TRAINING: DESCRIPTION OF LESSONS

Robert F. Yeager, Champaign, Illinois

Zita M. Simutis
U.S. Army Research Institute for the Behavioral and Social Sciences

EDUCATIONAL TECHNOLOGY AND SIMULATION
TECHNICAL AREA

FOREWORD

The Embedded Training and Gaming Team of the Army Research Institute (ARI) performs research in the area of computer-based simulations and gaming (both instructional and wargaming) methods with applicability to military training. A work unit of this team is concerned with improving training via instructional gaming techniques.

Instructional games are commonly used in the educational community, yet little research concerned with the educational effectiveness of game-based techniques has been conducted. This paper outlines a series of computer-based lessons on electronics maintenance which were developed for use in ARI's Game-Based Training Strategies Work Unit.

ARI research in this area is conducted under Army Project 2Q162722A764, FY 78 Work Program. The work reported here was accomplished by Mr. Robert F. Yeager, Champaign, Illinois, Contract No. MDA903-78-M-AA25 and personnel of the ARI Educational Technology and Training Simulation Technical Area, now called the Educational Technology and Simulation Technical Area.

JOSEPH ZEIDNER
Technical Director

COMPUTER-BASED ELECTRONICS MAINTENANCE TRAINING: DESCRIPTION OF LESSONS

BRIEF

Requirement:

To develop a series of computer-based lessons on electronics maintenance suitable for use in training logic circuit concepts within an experimental paradigm.

Procedures:

A series of computer-based lessons on electronics maintenance were designed and developed on the University of Illinois PLATO Computer-Based Education System. These lessons provide instruction and practice in reading logic symbols, logic diagrams, and electronic troubleshooting. Associated management lessons allow the experimenter/instructor to create individualized group sequences, either between or within lessons. Full data storage and retrieval mechanisms are also available.

Utilization:

These lessons are being used as a transfer task in a series of experiments on the training effectiveness of instructional games. The described lessons would also be suitable to supplement a course in electronics maintenance training.

TABLE OF CONTENTS

DESCRIPTION OF INSTRUCTIONAL LESSONS

Description of Elements.

This section describes the instructional and management lessons which were developed to train reading of logic circuit symbols, logic diagrams, and troubleshooting. Each description includes the following three elements:

General Description. The lesson is described and any special features of the lesson are listed. All lessons follow standard programming conventions which enable lessons to work with the PLATO system supported management system, currently called "mrouter." Specifically:

(1) each lesson sets a "lesson completed" flag to indicate whether or not the students have come to the end of that lesson;

(2) each lesson returns a "score" generally computed as a percent of correct answers the student made [100 * (right/total)]; and,

(3) each lesson sets "restart" flags which enable a student to leave the lesson, and, upon returning to the lesson at a later time, to restart at an appropriate section without having to start at the beginning of the lesson.

Parameters. "Parameters" are those variables which an experimenter/instructor can set in order to change the way the program operates. Experimenters/instructors can set parameters for individuals, for groups, or for all students using a lesson.

Data. Sections on data describe all the data that is currently collected. The data are currently saved in databases associated with the signon record. The data can be retrieved for individual students or it can be processed for groups. The following data are saved for each lesson:

(1) the name of the lesson;

(2) the exercise used;

(3) date lesson was used;

(4) amount of time spent in the lesson, accurate to seconds;

(5) whether or not the student completed the lesson;

(6) score, stated in terms of a percent (right/total);

(7) total number right;

(8) total trials.

INSTRUCTIONAL LESSONS

General Orientation Instructional Lesson (-lcintro-).

General Description. This lesson consists of two parts: (1) a general orientation to the experiment; and, (2) a biographic information survey.

The biographic information collected from the student includes:

(1) age, in years;

(2) time in service, in months;

(3) rank, saved in a ten character alphanumeric string;

(4) time in grade, in months;

(5) G.T., saved in a ten character alphanumeric string;

(6) MOS, saved as a ten character alphanumeric string;

(7) time in MOS, in months;

(8) education level, by grade level.

The student is also asked whether or not he is familiar with (a) computers, (b) electronics, or (c) auto-mechanics. If he indicates that he has had experience in any of those areas, he is further asked whether he has maintained or repaired equipment in the area.

Finally, the student is asked whether he is familiar with any of sixteen common games. From the same list, the student is asked to mark the games he has played; and, from the latter list, the student is told to rank the games he has played from favorite to least liked game.

Finally, the student is shown a summary page listing all his answers (Figure 1). The student may return from the summary page to any part of the lesson and change an answer.

Parameters. None.

Data. The data are stored permanently and associated with the signon record; it has been assumed that the student would not sign on with his real name so that there would be no way to associate the biographical data with a particular individual.

The data are stored in a format which will allow the experimenter to reconstruct the summary page for each student. The data may also be

processed automatically so that specific items can be correlated with other data.

Logic Diagram Instructional Lesson (-lcinstruct-).

General Description. This is a tutorial program to train a student on the interpretation of logic circuit symbols and diagrams.

In the tutorial, the student is first taught the meaning of the AND symbol; he steps through an input-output table which shows all four possible inputs and outputs (Figure 2). Then he is quizzed on his knowledge of the AND symbol (Figure 3).

In this quiz, the student is shown two inputs to an AND gate and asked to determine the output. He is given three problems; the problems are randomly generated, but a check is made to see that each student is given each possible type of problem (e.g., students will always see 1 + 1 and 0 + 0; they may see 1 + 0 or 0 + 1).

The student is recycled through the instruction if he gets two or more wrong on the quiz; but he is allowed to continue to the next section after two repeats.

After the section on AND, the student proceeds through sections on OR, NAND, and NOR. After the sections on OR and NOR, there are brief reviews.

Next, the student is shown the meaning of the INVERT symbol.

Then he is given a test with twelve simplified diagrams with mixed symbols (Figure 4); he must match the output values for each diagram. Incorrect answers receive explanatory feedback (Figure 5).

Finally, the student is given a brief demonstration of NODES and crossing lines.

Parameters. None.

Data. The score is based on the student's performance in the twelve item test.

The lesson also saves student performance on each of the four quizzes on AND, OR, NAND, and NOR; and on the twelve item test. For each of these, each item the student sees is marked; and each item the student got correct is marked. Furthermore, for the test only, the sequence of items is kept.

Fault Isolation Instructional Lesson (-lcfault-).

General Description. This is a tutorial lesson designed to teach fault isolation procedures.

The lesson starts by showing the student a mock-up of a computer circuit board (Figure 6); and it tells him that the objective is to determine if there is a malfunction by comparing the lights on the board to a circuit diagram.

The lesson explains how the lights on the circuit board operate (Figure 7); how to relate the lights on the circuit board to symbols on a diagram (all provided on the screen, Figures 8 and 9); and how to determine if there is a malfunction by comparing expected outputs from the diagram with lights on the circuit board (Figure 10).

Finally, the student is given a short quiz of twelve items to determine if he can identify malfunctions (Figure 11). All diagrams used in the quiz are drawn on the screen; and they are in the simplest form. Incorrect answers receive corrective feedback (Figure 12).

Parameters. None.

Data. The student's score is based on performance in the twelve item quiz.

Logic Diagrams Practice Lesson (-lcprac-).

General Description. This lesson gives a student practice solving circuit diagrams. The lesson starts with a page describing the task to the student (in Appendix 1). The student is then given diagrams and required to mark specified output values for each diagram (Figure 13).

The circuit diagrams used in the practice session were predetermined. All diagrams have been entered into a PLATO database. For each diagram, the student is shown the input values and asked to specify one or more output values. At the bottom of the screen there are boxes with "0" and "1" in them; the student may either touch the appropriate box, or can type the number; then he is to touch the output he wishes to mark; the number that is currently specified is written next to the output (Figure 14).

A scratch pad technique is available which allows a student to mark individual gates in the same way they mark the outputs; this allows a student to work through a diagram without having to remember the values of all the gates.

There is also a TABLES box at the bottom of the screen. If the student touches the TABLES box, and then touches an AND, OR, NAND, NOR, or INVERT symbol, he will be shown the possible inputs and outputs for that symbol.

In the lower right hand corner of the screen, there is a DONE box. When the student has completed marking the diagram, he touches the DONE box to have it judged. Incorrectly marked outputs are crossed out, and the student is told how many he had wrong (Figure 15).

Parameters. Using an exercise editor, the experimenter/instructor can set any of the following criteria for both the practice session as a whole, or for a list of diagrams within the practice session:

(1) amount of time allowed; when this time period expires, the student may finish working on the current diagram, but may not start a new diagram;

(2) maximum amount of time allowed; when the time period expires, the student will be "ripped out" of whatever he is currently doing, and the lesson will be terminated;

(3) number of diagrams required; the lesson ends either when the student completes the specified number of diagrams, or when he runs out of diagrams;

(4) success criteria; the lesson will end when any of the following criteria are met;

(a) total correct;

(b) consecutive correct after a specified number of problems;

(c) percent correct after a specified number of problems;

(5) failure criteria: the lesson ends if any of these criteria are met; the criteria are the same as those above, except that wrong answers are counted instead of correct answers.

None of the above criteria need to be set; and any combination of criteria can be used.

If the criteria are attached to a subset of diagrams, then matching a criteria can allow branching to another subset of diagrams. Thus the experimenter/instructor can establish levels of difficulty, and specify the algorithms for moving to a different subset of diagrams.

The experimenter/instructor can specify the diagrams in any order; or the diagrams can be given in random order for each student.

For each diagram, the input values may be specified directly, allowed to remain at their default setting, or be reassigned randomly for each student. If the input values are changed, all values for all gates and outputs are re-computed; and the changes only affect the listing of that particular diagram. Therefore, it is possible to use the same diagram more than once, but with different input values.

Each diagram may also have its required outputs set; the list of required outputs may be specified directly, left at their default setting, or assigned randomly for each student.

Data. The student's score is based on the total number of diagrams which had all outputs judged correct. Therefore, if a student gets three outputs correct, but one wrong, the entire diagram is marked as wrong for the purpose of scoring. Total number of correct outputs is also stored in a database.

The following data are currently stored for each diagram:

(1) name of the diagram;

(2) time spent on the diagram;

(3) the correct values of the outputs;

(4) the required values of each output as assigned by the student.

Data are stored by signon records and can be retrieved for individual or group analysis.

Logic Diagrams Test Lesson (-lctest-).

General Description. This lesson is identical to the lesson described in the previous section, except that:

(1) different directions are given (see Appendix 2);

(2) the TABLES option is not available (Figure 16);

(3) the student is not told whether or not he has made errors; he is simply routed to the next diagram (Figure 17).

Parameters. The parameters for the test are identical to the parameters described in the previous lesson.

Data. The data saved for this lesson are identical to the data saved for the previous lesson.

Fault Isolation Test Lesson (-lcfaultest-).

General Description. This packet consists of thirty-two fault isolation problems specially designed for this test. In addition, any of the diagrams used in the Logic Diagram Practice Session of the Test may be used in this lesson; therefore, almost one hundred diagrams are available; and all of them can be used many times by changing their values and/or location of a malfunction.

All circuit diagrams are in hard copy so that they can be used with this lesson.

In this lesson, the students are shown a circuit board with lights; lights which are "on" indicate a value of "1" at that location; lights which are "off" are "0" value. At the top of the screen, the name of the diagram and an identifying page number are written; the student uses the hard copy diagram describing the circuit. Then, using the indicator lights, the student must determine if the circuit is operating correctly. A student can mark the values of the gates on their copies of the circuit diagrams.

The student indicates a malfunction by touching the indicator light; the symbol above the indicator light is highlighted (Figure 18). An answer may be changed by touching the light a second time, or by selecting a different answer.

If there is no malfunction, the student touches a NO MALFUNCTION box at the lower left corner of the screen (Figure 19).

When the student is sure he is correct, he touches a DONE box. He is told whether or not he is correct, and is given remedial feedback if wrong. The student must successfully solve a problem before continuing to the next one (Figure 20).

Parameters. Parameters are similar to those described for Logic Diagrams Practice Lesson. However, for this lesson, it is not necessary to indicate "required outputs." Instead, the experimenter/instructor may specify the type of malfunction for each diagram; the options are:

(1) default setting; each diagram is preset with a specified malfunction gate; approximately twenty percent of all diagrams are specified as having no malfunction;

(2) no malfunction;

(3) malfunction at a specified gate number; the experimenter selects the gate;

(4) a definite malfunction, but at a random gate;

(5) a malfunction according to the probabilities preset for the list of diagrams. For example, the experimenter may specify that every four out of five diagrams presented should have a malfunction.

Data. Data are similar to that described for the Logic Diagrams Practice Lesson except that a list of gates the student marked as having malfunctions is kept instead of the values the student assigned to each gate.

MANAGEMENT LESSONS

The following management programs were designed to facilitate the use of the lessons and the editing of the data bases associated with the lessons.

Index of Lessons (-lcindex-). Lessons needed by the experimenter/instructor can be accessed from this lesson. The index could be used as an "instructor router," if desired.

Exercise Editor (-lcexerl-). This editor allows the creation and editing of exercises for (a) practice sessions; (b) tests on circuit diagrams; and (c) tests on finding malfunctions in circuits.

For each exercise, the instructor can set criteria (Figure 21); and enter "lists" of diagrams, each with its own unique set of criteria (Figure 22). An exercise may contain up to twenty-six lists; each list may contain up to twenty-six diagrams (Figure 23).

Each diagram may be edited so that inputs, required outputs, and malfunctions can be changed (Figure 24).

Circuit Diagrams Editor (-lcedit-). In order to reliably and compactly describe the circuit diagrams, an editor was developed. The diagram is described in "lines" of data (Figure 25); a typical diagram takes about one hundred "lines" to describe.

For gates, each line of data contains information about:

(1) gate numbers; all gates are numbered by their location on the screen;

(2) gate numbers of input gates; thus, the value of a gate can be determined by computing it from the values of its input gates;

(3) the screen location of the symbol.

For all other display items, the "line" contains the type of display (box, line, or vector), and the screen locations.

The editor can display the symbols with their gate numbers, and the gate numbers of the inputs to each gate (Figure 26); this allows verification of each diagram.

The editor also edits the data base which associates touch locations on the screen with gate numbers (Figures 27 and 28).

All data on the diagrams are stored in a permanent data base; this editor allows permanent changes in the description and values of a circuit diagram.

APPENDIX 1: DIRECTIONS FOR THE CIRCUIT DIAGRAMS PRACTICE SESSION

Logic Diagrams Practice Session

In this lesson you will practice marking logic diagrams.

You must mark each output which is marked by a question mark.

You can mark an output by touching a "0" or "1" box at the bottom of the page, and then touching the output you wish to mark.

When you have marked all the required outputs touch the DONE box.

You may also mark any of the other gates on the page in order to trace a circuit. You will only by judged, however, on whether or not the required outputs are marked correctly.

There will also be a box called "TABLES" at the bottom of the screen. If you touch that box, and then touch a gate, you will be shown the truth table for that gate.

You can also press the HELP key at any time for additional options.

APPENDIX 2: DIRECTIONS FOR THE CIRCUIT DIAGRAMS TEST

Logic Diagrams Test

This is a test to see how well you can read logic circuits. It works similarly to the practice session except that you cannot ask to see the TABLES. Also, you will not be told whether you are correct or incorrect.

You must mark each output which is marked by "?".

You can mark an output by touching a "0" or "1" box at the bottom of the page, and then touching the output you wish to mark.

When you have marked all the required outputs touch the DONE box.

You may also mark any of the other gates on the page in order to trace a circuit. You will only be judged, however, on whether or not the required outputs are marked correctly.

You can also press the HELP key at any time for additional options.

APPENDIX 3. The PLATO Computer-Based Instructional System

PLATO[1] is an interactive, dedicated computer-based instructional system developed at the Computer-Based Education Research Laboratory (CERL) at the University of Illinois at Urbana-Champaign. PLATO systems have been installed at several US and overseas locations. Each PLATO system consists of up to about 1000 terminals which are connected via telephone communication lines to a large central computer. The materials described in this report were developed on the CERL PLATO system and are available to all CERL users. Users on other PLATO systems may, upon request, gain access to the courseware. A PLATO terminal provides the interface between the user and the central computer. A terminal's major components are a keyset which transmits users requests or inputs from the terminal and a visual display panel for presenting lesson material. Touch panel input capabilities are also available.

Additional information about the PLATO system can be obtained from Smith, S. and Sherwood, B. A., "Educational uses of the PLATO Computer System," Science, 1976, 192, 344-352.

---

[1]Commercial designations are used only for precision of description. Their use does not constitute indorsement by the Department of the Army or the U.S. Army Research Institute.

The following is a summary of all the answers you made. You may change an answer by typing the appropriate page number. If all the answers are correct, press NEXT to continue.

Page 1

Age	= 21 years	Time in Service	= 6 months
Rank	= Skipped	Time in Grade	= Skipped
MOS	= Skipped	Time in MOS	= Skipped
GT	= Skipped	Education level	= Grade 12

	Familiar	Maintained	Repaired
Page 2: Computers	Little bit	No	No
Page 3: Electronics	Not at all	-	-
Page 4: Automobile Mechanics	Very	Yes	Yes

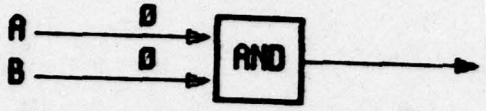
Pages 5 + 6	**Familiar**	**Played**
Backgammon	**Yes**	**Yes**
Battleship	**No**	**No**
Blackjack	**Yes**	**No**
Bridge	**Yes**	**Yes**
Chess	**No**	**No**
Checkers	**Yes**	**Yes**
Gin Rummy	**Yes**	**Yes**
Go	**No**	**No**
Hearts	**No**	**No**
Masternind	**No**	**No**
Monopoly	**No**	**No**
Parchesi	**No**	**No**
Poker	**Yes**	**Yes**
Stratego	**No**	**No**
Tic-Tac-Toe	**Yes**	**Yes**
Uiff 'N Proof	**No**	**No**

Page 7: Ranking of Games Played

1. **Checkers**
2. **Gin Rummy**
3. **Backgammon**
4. **Bridge**
5. **Poker**
6. **Tic-Tac-Toe**

Figure 1. Lesson -lcintro-. Summary page showing student all of his answers and allowing him to return to any section to change an answer.

To illustrate:



INPUT		OUTPUT
A	B	X = A B
1	1	1
0	1	0
1	0	0

Keep pressing **NEXT** to complete the **AND** table.

Figure 2. Lesson -lcinstruct-. The student presses the NEXT key to complete a table of inputs and outputs for the AND symbol.

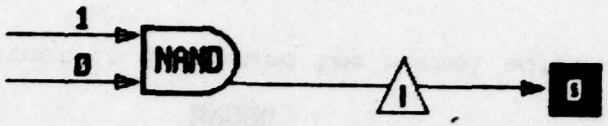
Problem 1 of 12

Now you determine the output for these examples:

Touch either 0 or 1 (you can also type the number).

Figure 3. Lesson -1cinstruct-. Quiz on the AND symbol; the student must touch or type the number "1".

Problem 9 of 12

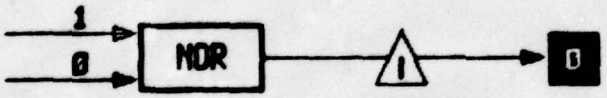


RIGHT

Press Next

0 1

Figure 4. Lesson -lcinstruct-. Mixed practice test at the end of the lesson; the student touches "0" and are marked "RIGHT".



WRONG

The correct output is **1** because the normal output of **0** is **INVERTED.**

Press Next

Figure 5. Lesson -lcinstruct-. Mixed practice test; in this case, the student has gotten the question wrong and receives explanatory feedback.

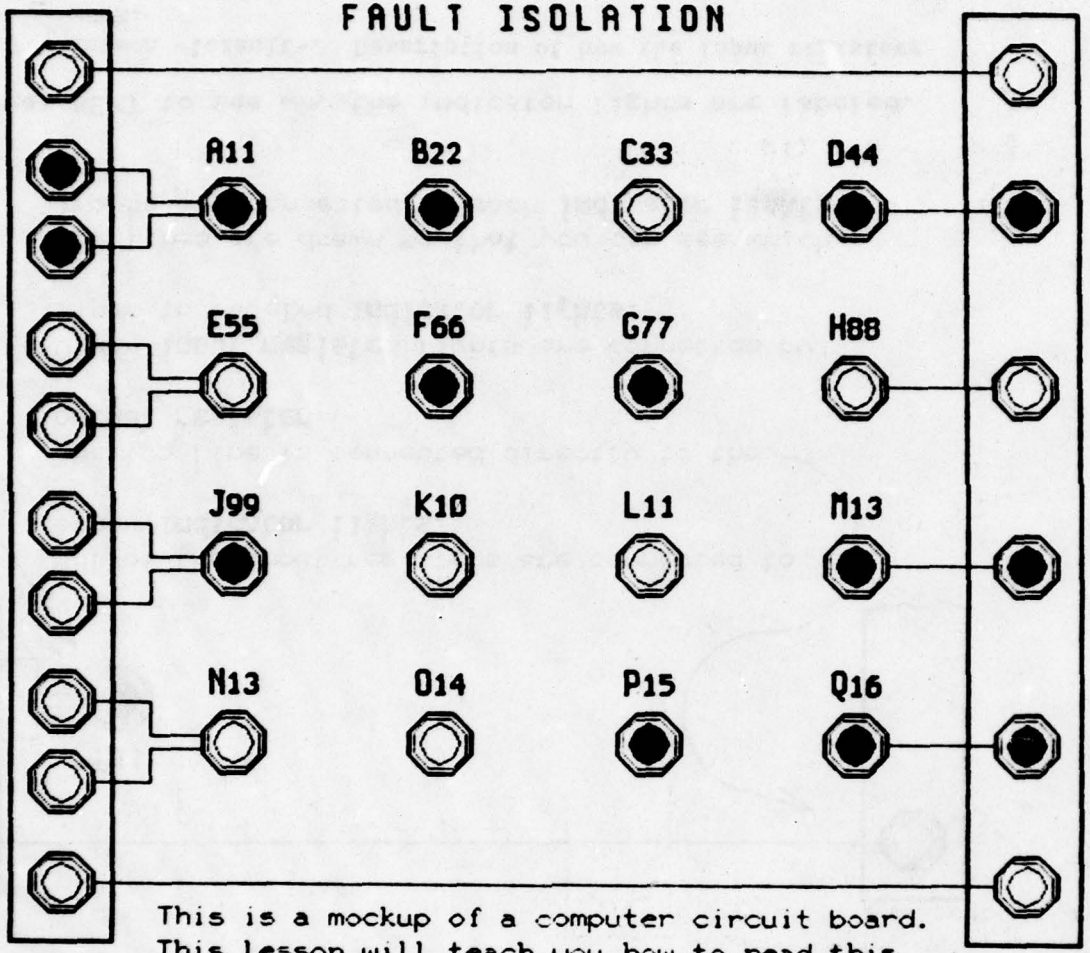


Figure 6. Lesson -lcfault-. First page of the lesson on finding malfunctions in logic circuits.

Symbol Location

All of the input registers are connected to other **indicator lights.**

The top line is connected directly to the **output register.**

Other **input register** lights are connected by lines to labeled **indicator lights.**

The lines are drawn so that you can see which inputs are connected to each **indicator light.**

Press NEXT to see why the indicator lights are labeled.

Figure 7. Lesson -lcfault-. Description of how the input registers work.

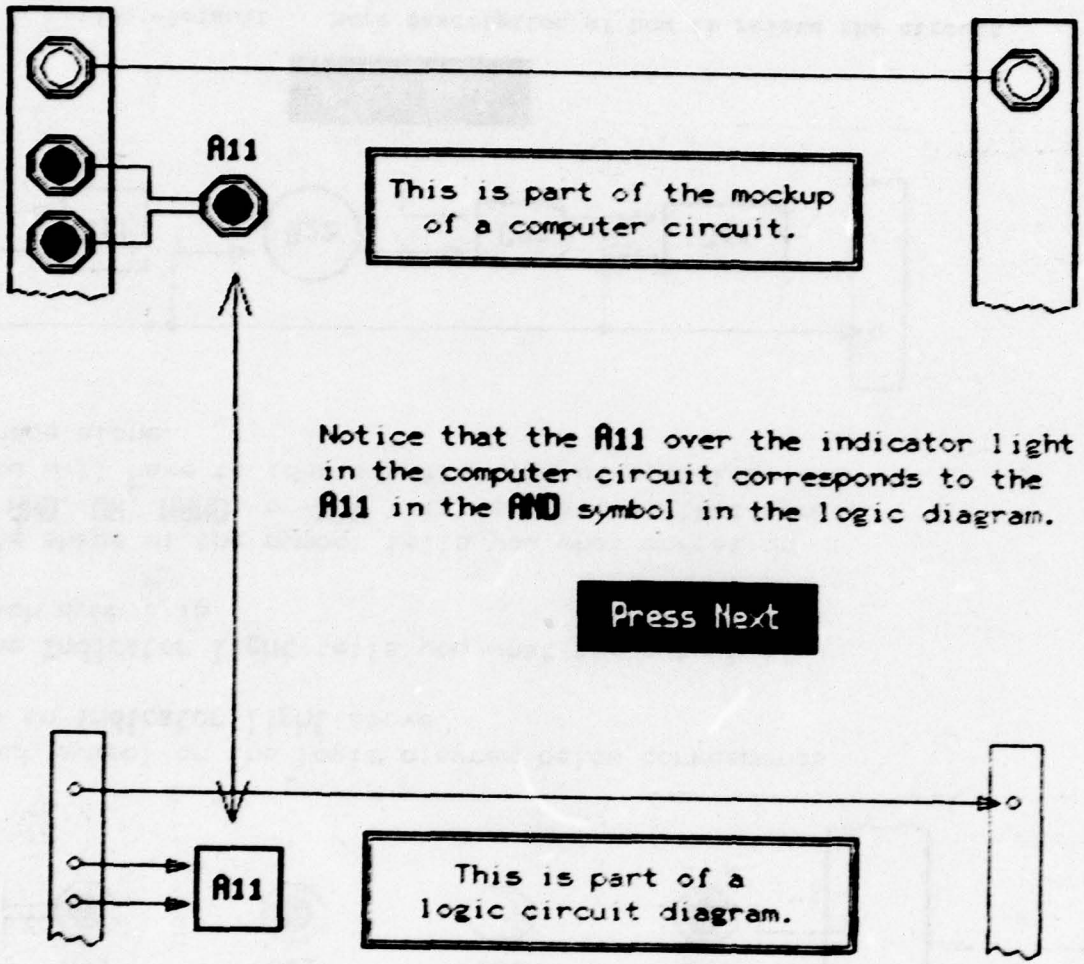
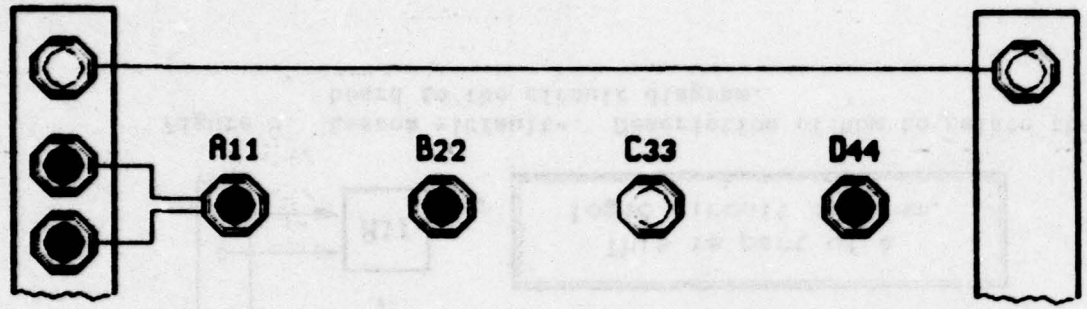


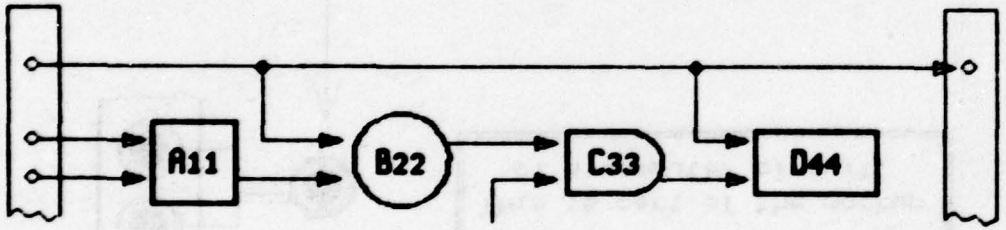
Figure 8. Lesson -lcfault-. Description of how to relate the circuit board to the circuit diagram.



Each symbol on the logic diagram below corresponds to an **indicator light** above.

The **indicator light** tells you what the output of each symbol is.

The shape of the symbol tells you what operation (**AND**, **OR**, **NAND**, or **NOR**) takes place at that symbol. You will have to identify the type of symbol by its shape alone.



Press Next

Figure 9. Lesson -lcfault-. More description of how to relate the circuit board to the circuit diagram.

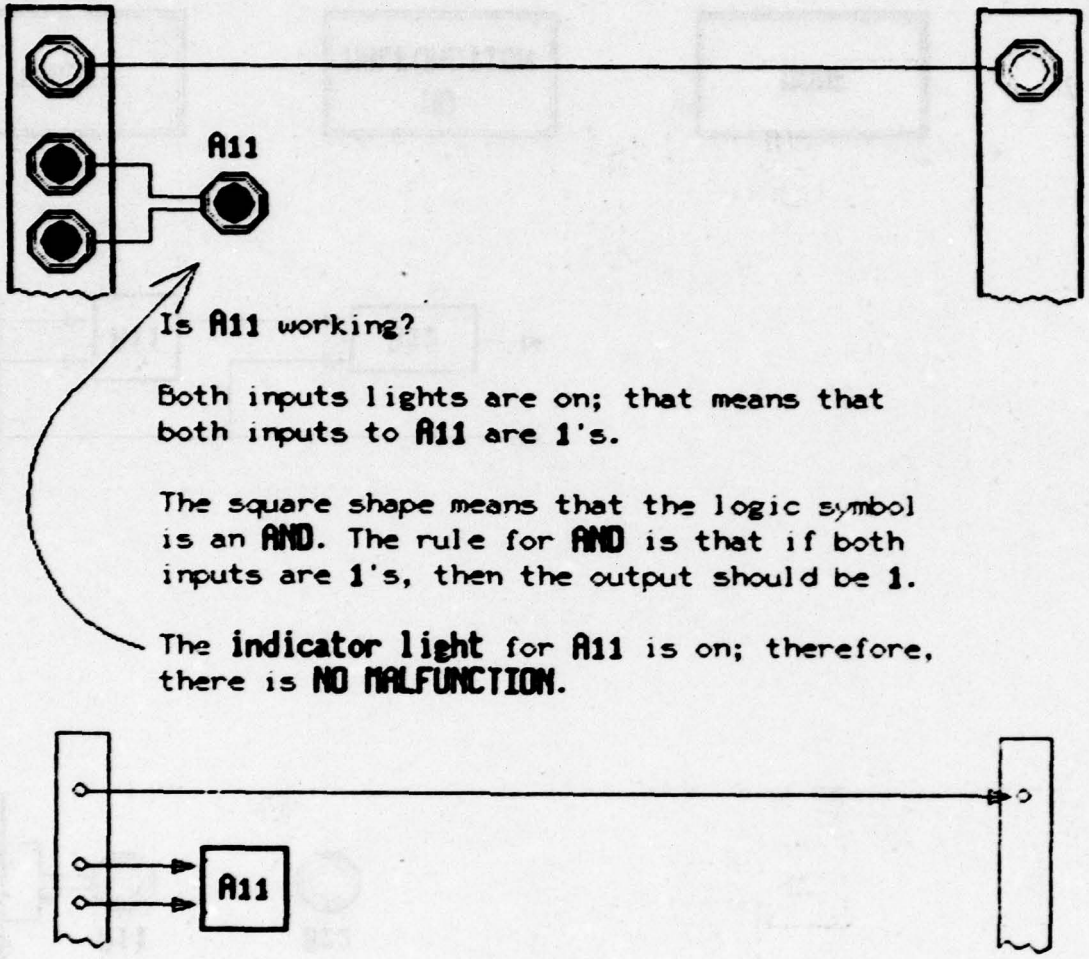


Figure 10. Lesson -lcfault-. How to determine whether or not there is a malfunction in a circuit.

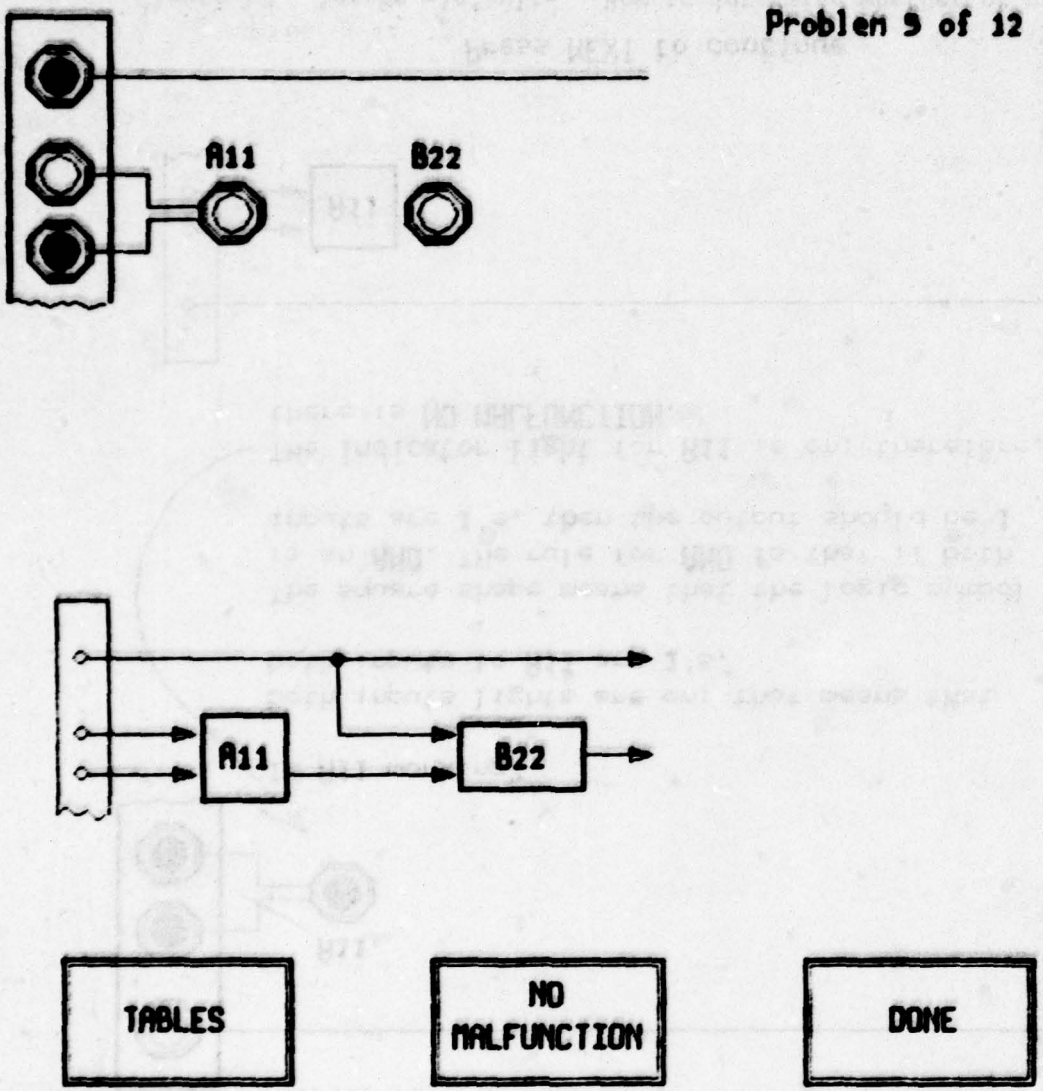
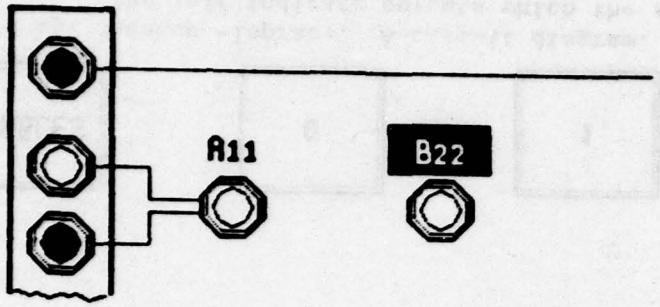


Figure 11. Lesson -lcfault-. Test on whether students can determine if there is a malfunction.



WRONG. There is no malfunction.

The inputs to the **NOR** are **1** and **0**; therefore, the output should be **0**, and **B22** should be **OFF**.

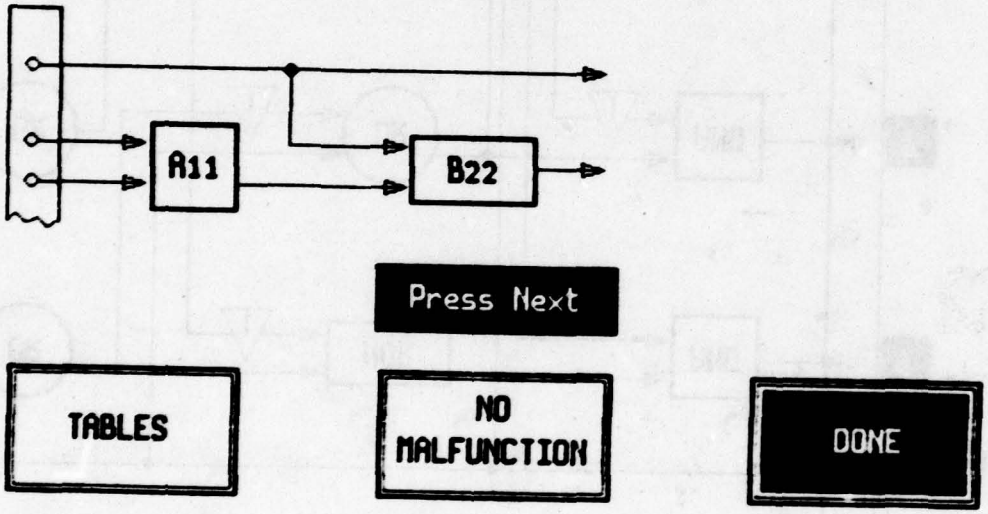


Figure 12. Lesson -lcfault-. Corrective feedback for test.

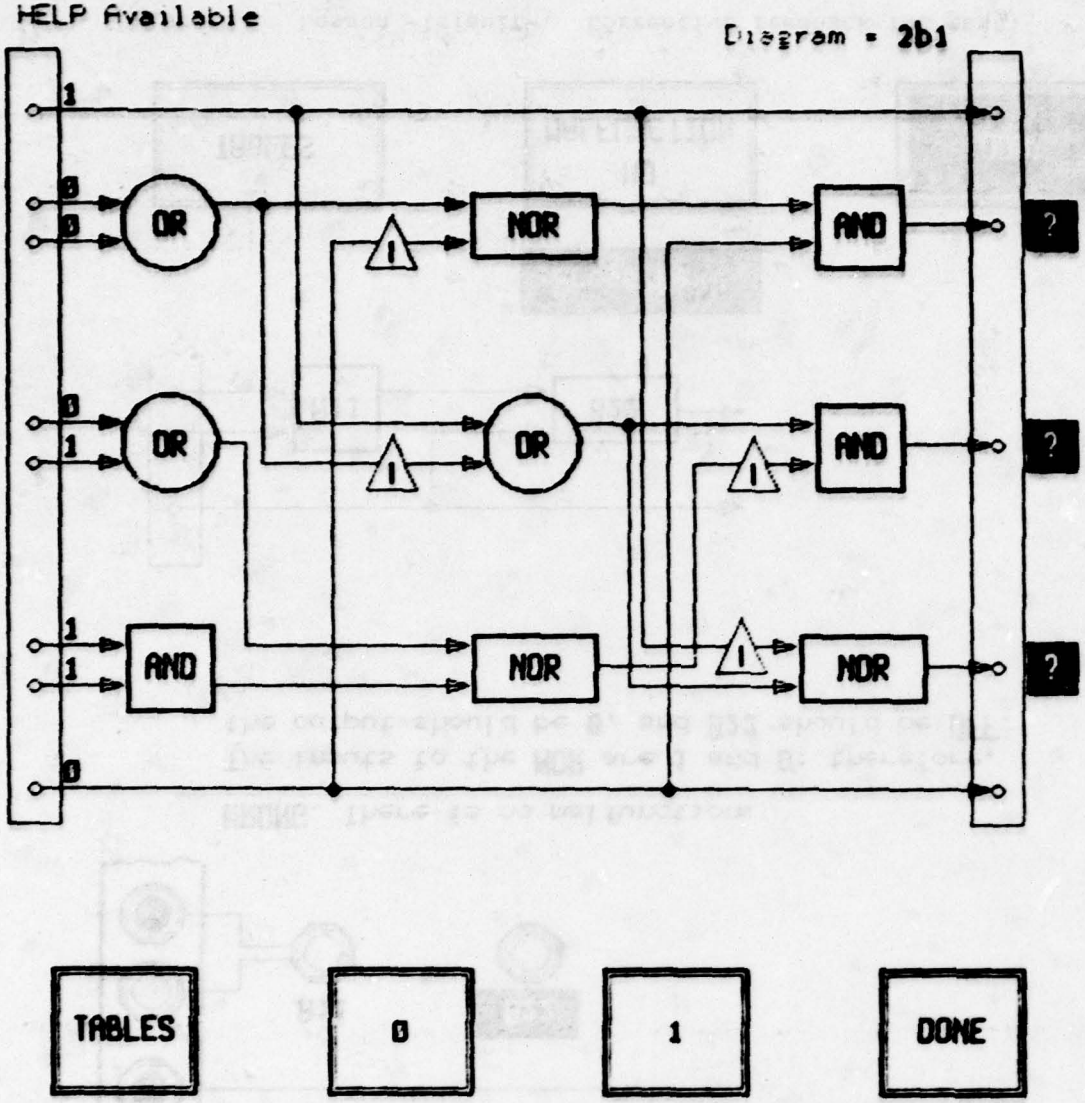


Figure 13. Lesson -lcprac-. A circuit diagram. The question marks at the left indicate outputs which the students must solve. Touching the TABLES box and then touching a symbol will show all possible values for that symbol.

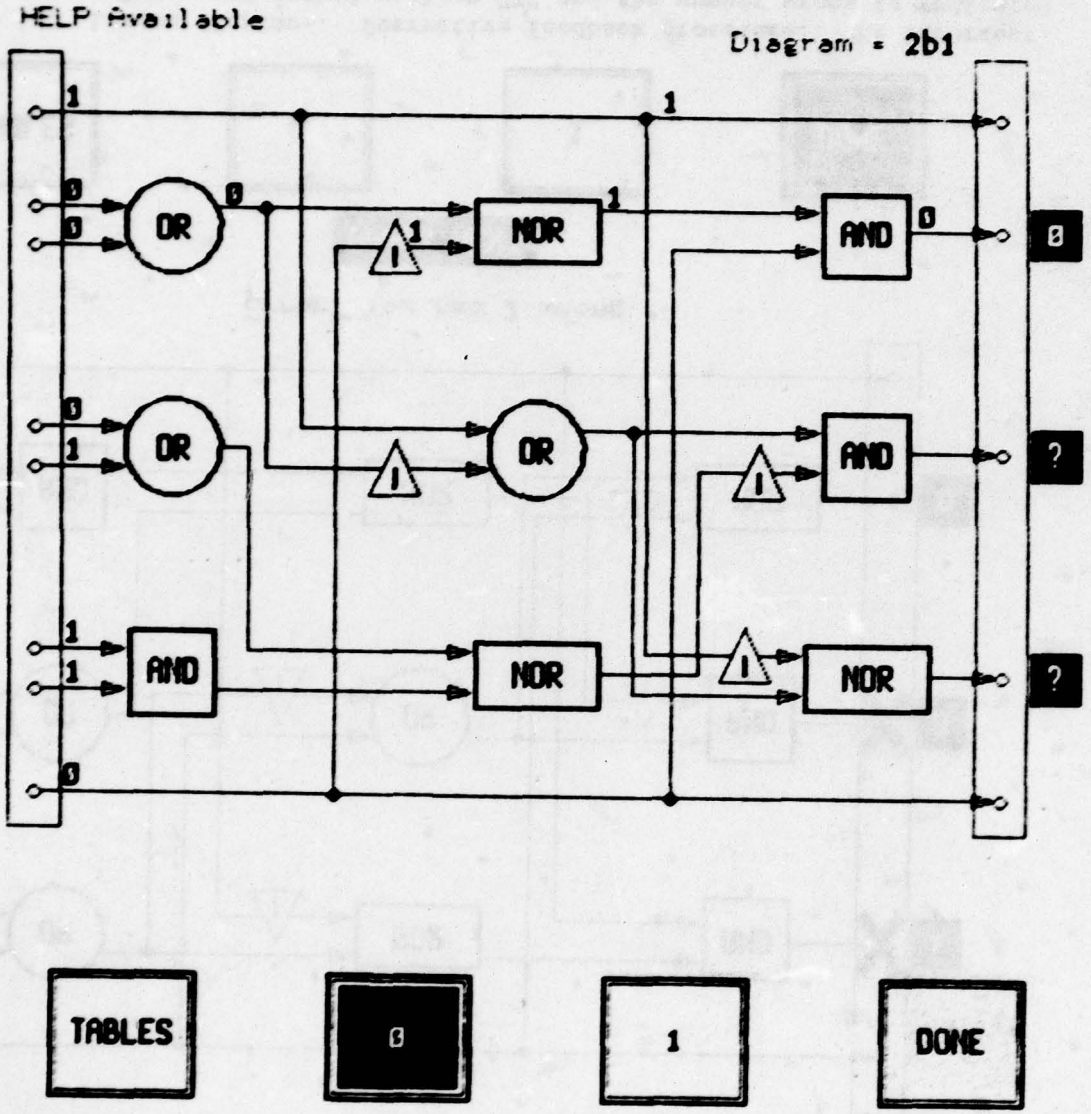


Figure 14. Lesson -lcprac-. The student may mark anyplace on the circuit diagram by touching "0" or "1" at the bottom, and then touching the gate he wishes to mark. In this example, the "0" is covered meaning that each gate the student touches will be marked with a "0".

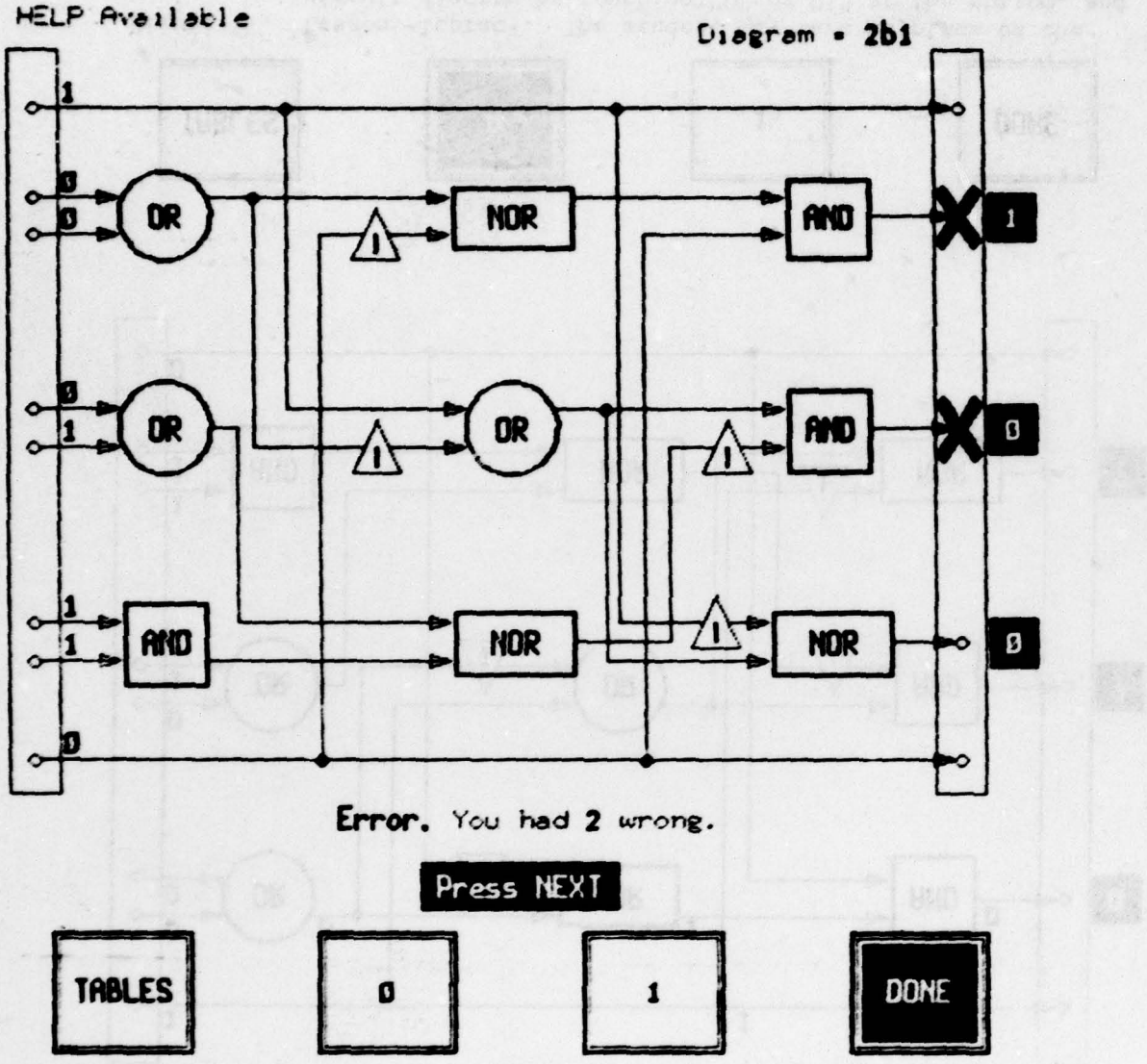


Figure 15. Lesson -lcprac-. Corrective feedback procedure: the incorrect outputs are marked with an "X" and the number wrong is indicated at the bottom.

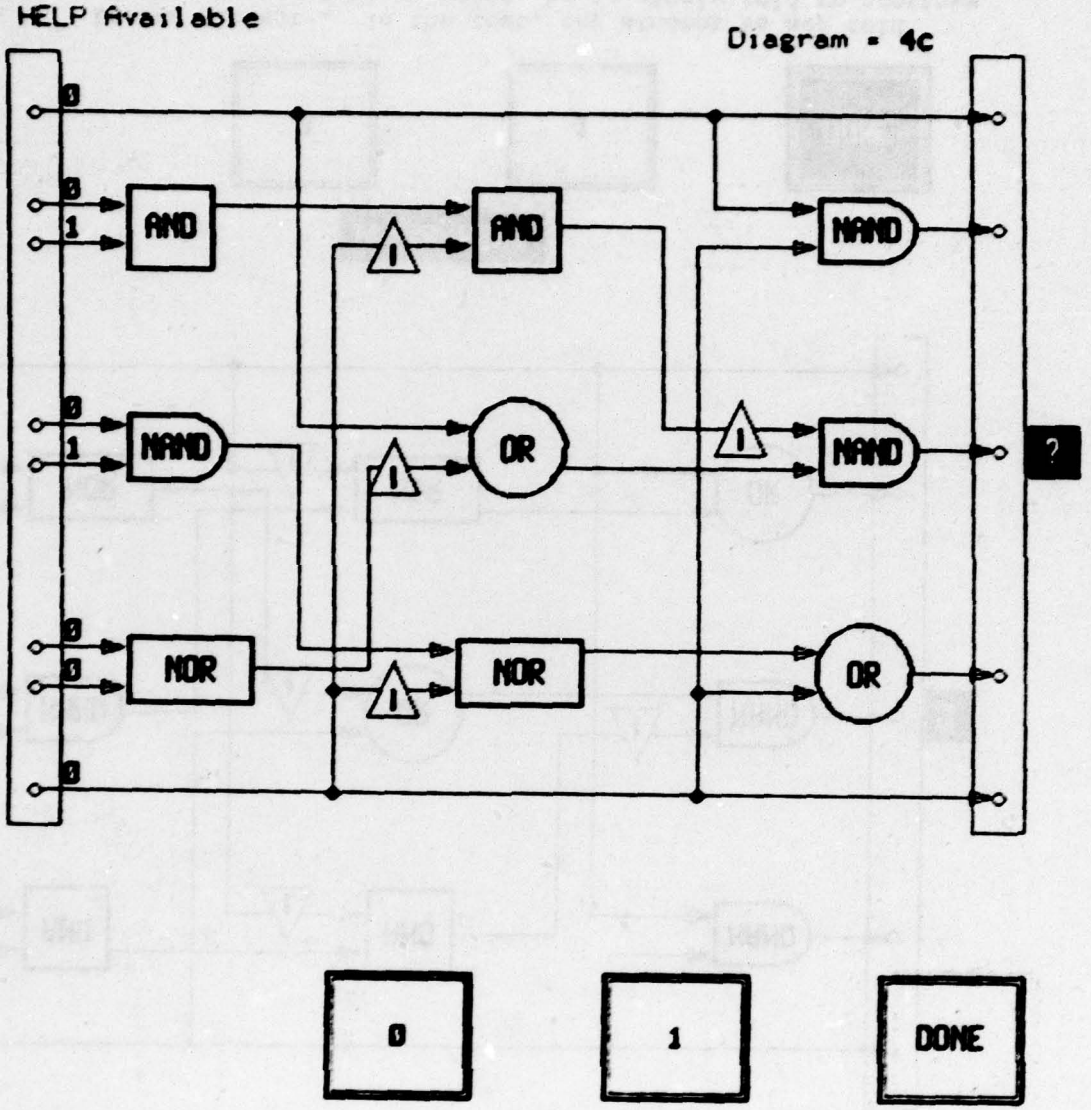


Figure 16. Lesson -lctest-. Circuit diagrams test: the lesson is identical to -lcprac- except that it does not allow the student to see the TABLES.

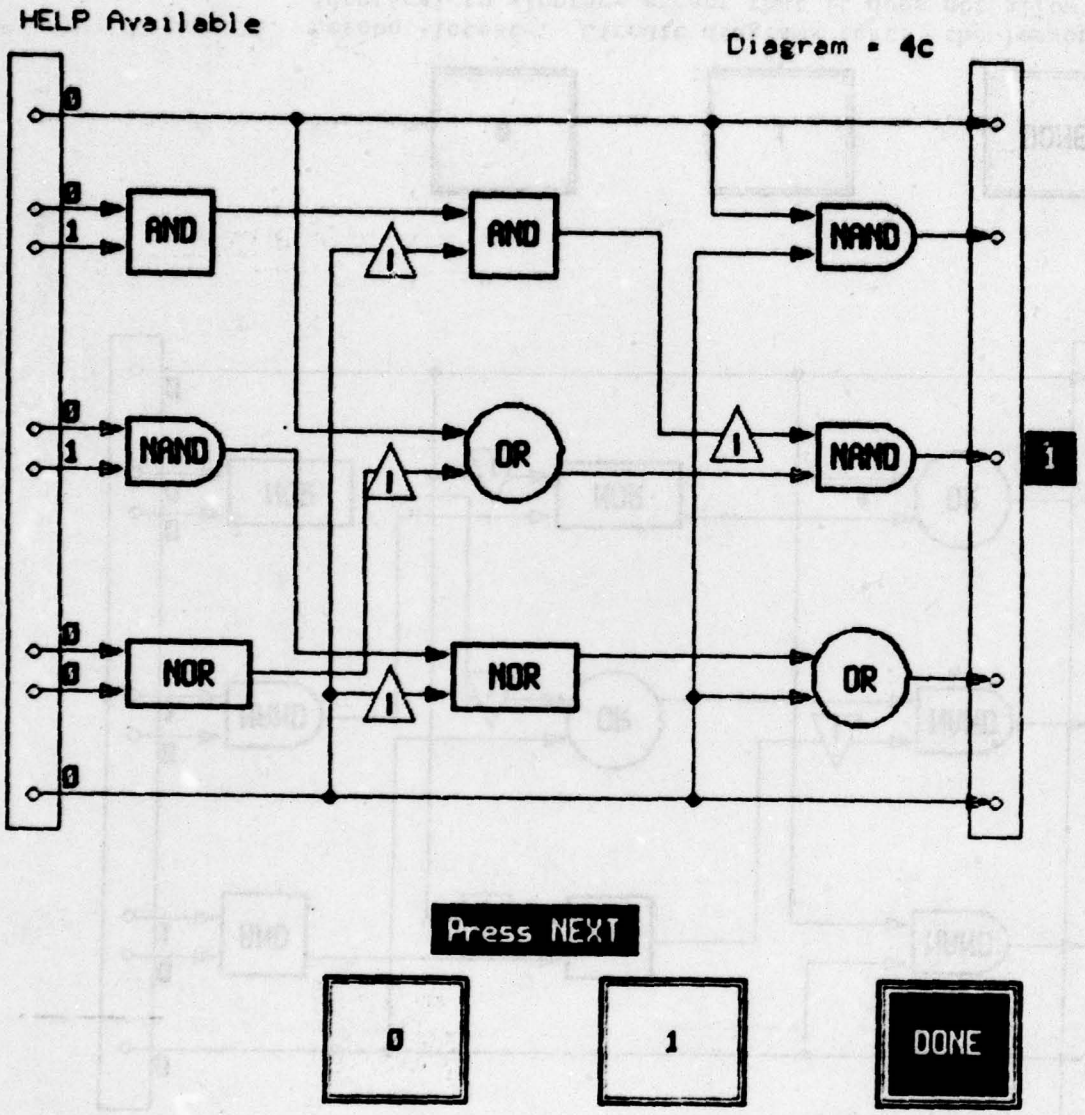


Figure 17. Lesson -lctest-. In the test, the student is not told whether he is right or wrong; he is simply told to continue to the next diagram.

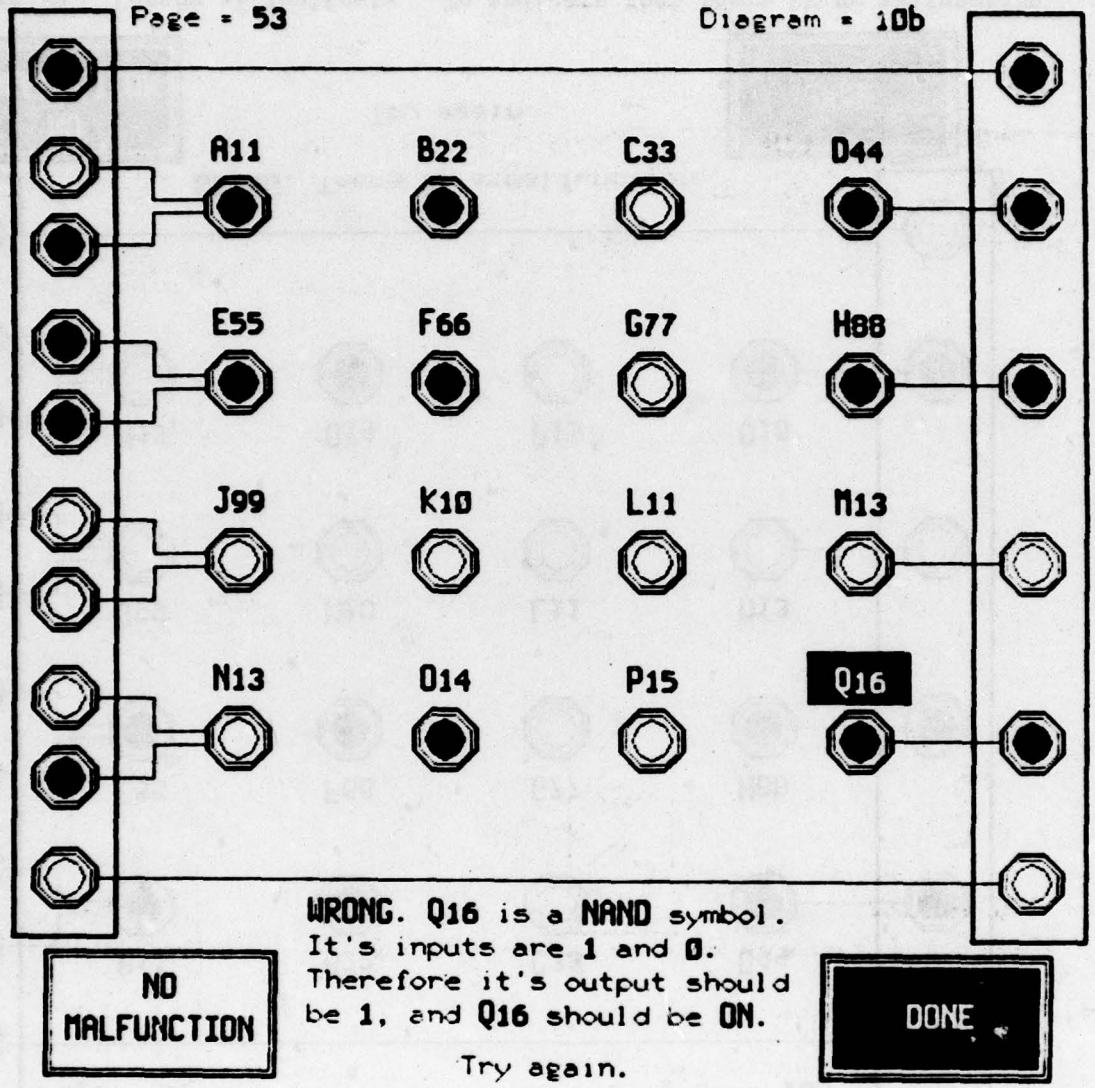


Figure 18. Lesson -lcfaultest-. The student has touched indicator light "Q16" to indicate a malfunction at that gate; the feedback explains why "Q16" is not the correct answer, and tells the student to try again.

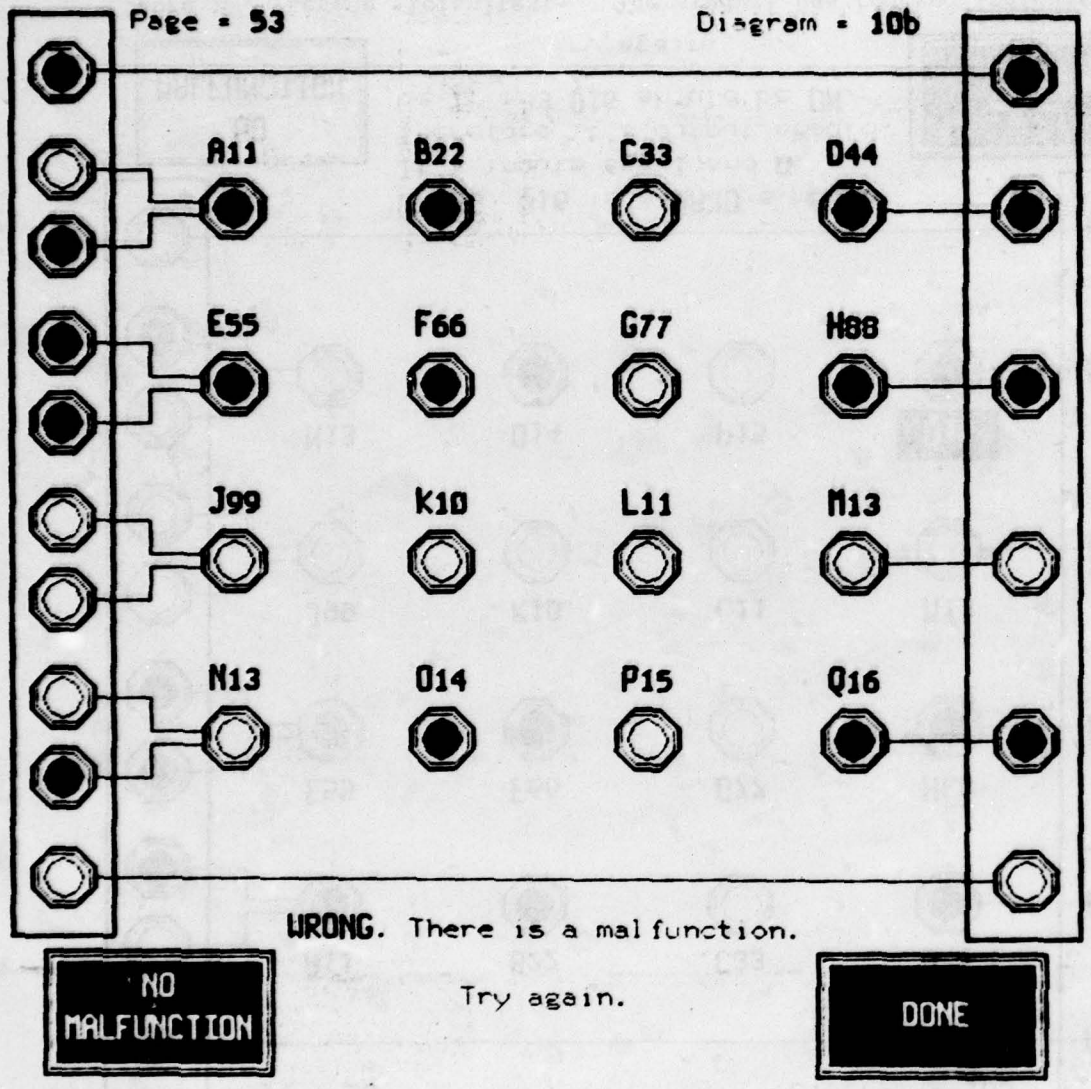


Figure 19. Lesson -lcfaultest-. To indicate that there is no malfunction in the circuit, the student can touch the "NO MALFUNCTION" box; in this example, there is a malfunction and the student must try again.

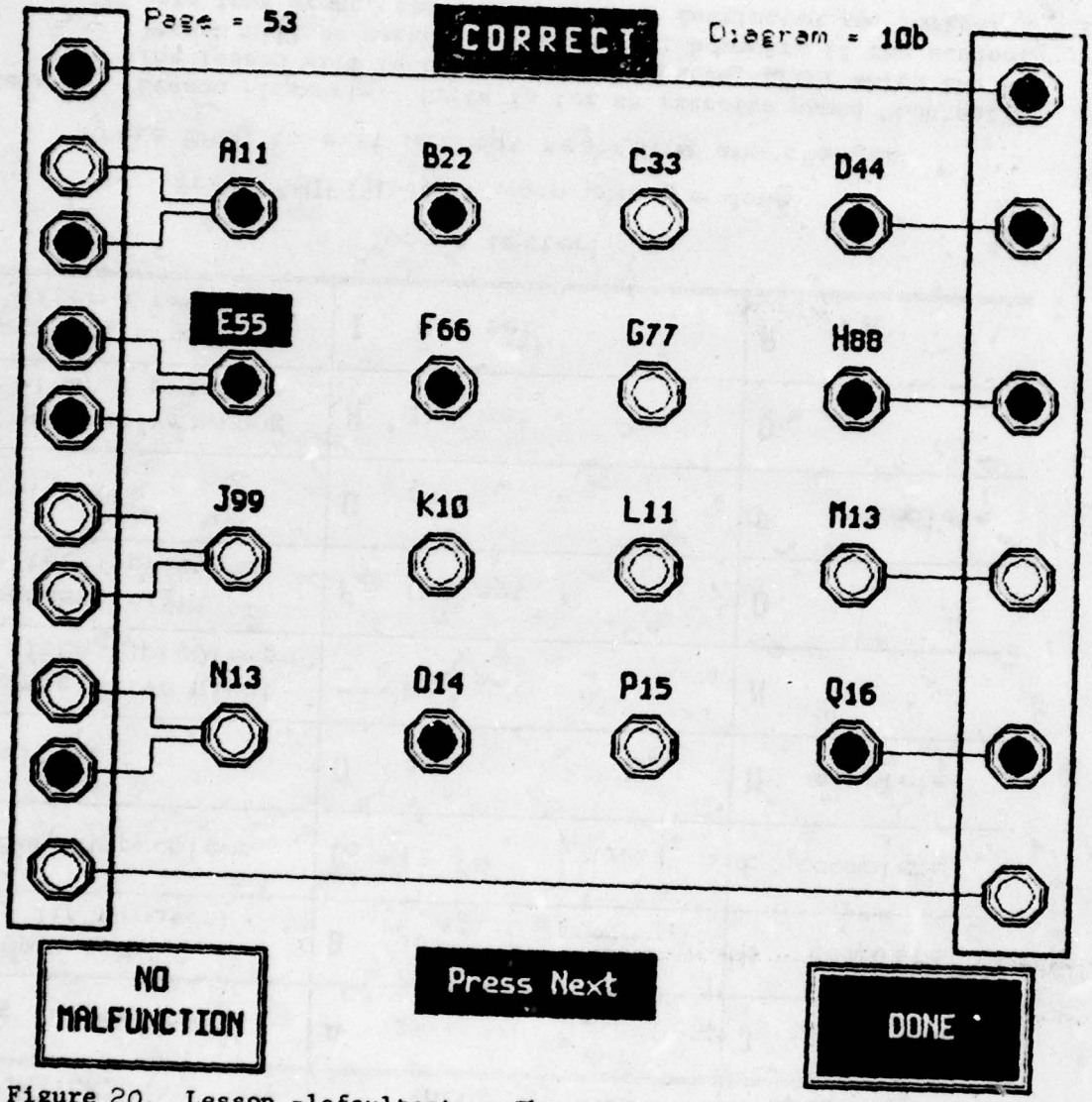


Figure 20. Lesson -lcfaultest-. The malfunctioning gate has been correctly identified as "E55". "CORRECT" flashes at the top, and the student can proceed to the next problem.

Set criteria for Practice Session "bobtest"

CRITERION	VALUE	LESSON STATUS
Time (in minutes)	A 25	J complete
Maximum time (in minutes)	B 30	K complete
Number of problems	C 12	L complete
Total right	D 8	M complete
X consecutive right after Y problems	E Not set	N
X percent right after Y problems	F Not set	O
Total wrong	G 4	P incomplete
X consecutive wrong after Y problems	H Not set	Q
X percent wrong after Y problems	I Not set	R

Type a letter

Press SHIFTED-NEXT when you are done

Press BACK to exit without recording any changes

Figure 21. Lesson -lcexerl-. Criteria for an exercise named "bobtest". The lesson will last twenty-five minutes, after which the lesson will be marked as "complete"; however, if the student gets four wrong, the lesson will be terminated and marked incomplete".

Criteria for list "List one", practice session "bobtest"

CRITERION	VALUE	BRANCH TO
Time (in minutes)	A 8	J List two
Maximum time (in minutes)	B 12	K List two
Number of problems	C 4	L List two
Total right	D 3	M List two
X consecutive right after Y problems	E Not set	N
X percent right after Y problems	F Not set	O
Total wrong	G 2	P Remedial1
X consecutive wrong after Y problems	H Not set	Q
X percent wrong after Y problems	I Not set	R

Type a letter

Press SHIFTED-NEXT when you are done

Press BACK to exit without recording any changes

Figure 22. Lesson -1cexer1-. Criteria for a list named "List one" within exercise "bobtest". The criteria are the same except that matching a criteria signals branching to another list instead of terminating the lesson (the lesson can also be terminated by branching to "complete" or "incomplete"). In this example, the student will go to "List two" after eight minutes; or to a list named "Remedial" after two wrong.

Edit list "List one" in practice session "bobtest"

1 Criteria for this list

2 Sequential / Random presentation (currently = sequential)

3 Delete a diagram

4 Change the name of this list

A 1a
B 2a
C 3a
D 4a
E 5a
F 10e
G

Type a number or a letter >

Press BACK to return to the practice session index

Figure 23. Lesson -lcexerl-. Options for a list of diagrams. A list may have up to twenty-six diagrams, lettered A through Z.

Edit diagram F, "1De"

A Set all inputs to "Default"

B Set all inputs to "Random"

C Set all outputs to "Default"

D Set all outputs to "Random"

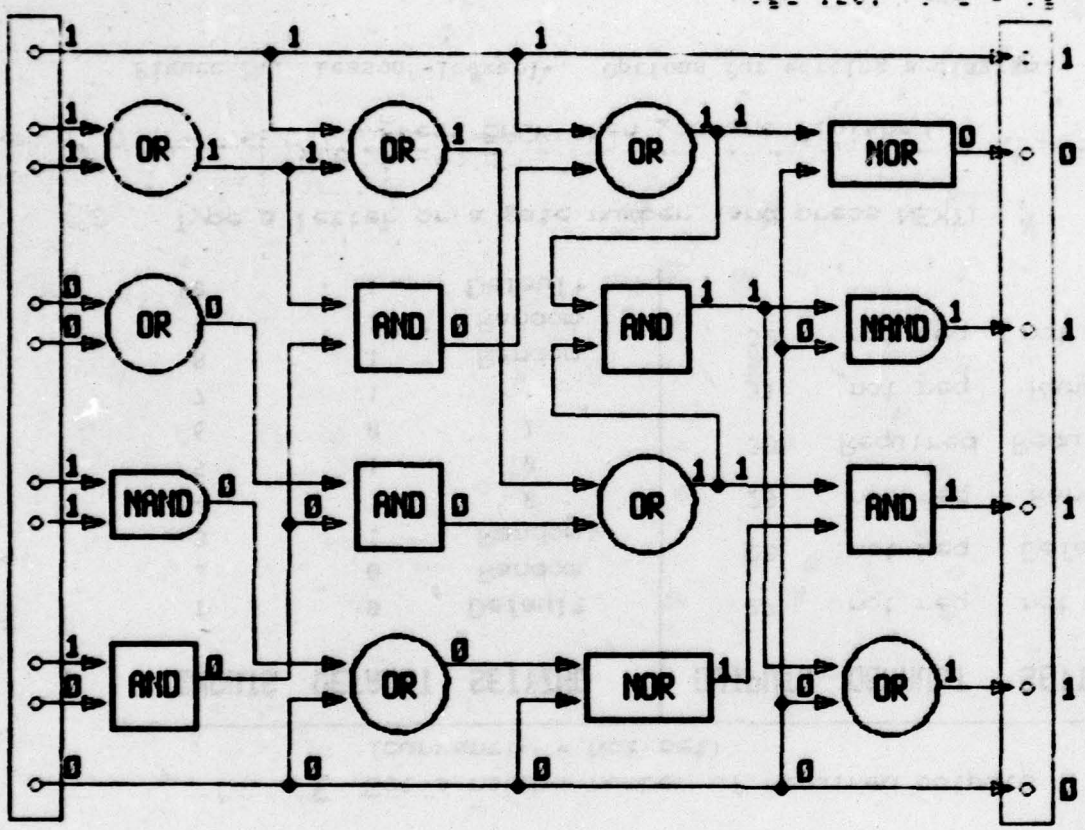
E Set a random number of required outputs
(currently = Not set)

INPUTS	DEFAULT	SETTING	OUTPUTS	DEFAULT	SETTING
1	Ø	Default	27	not req	not req
2	Ø	Random	28	not req	Default
3	1	Random	29	not req	Random
4	1	Ø	3Ø	Required	Required
5	1	Ø	31	not req	Random
6	Ø	1	32	not req	not req
7	1	1			
8	1	Random			
9	1	Random			
1Ø	1	Default			

Type a letter or a gate number (and press NEXT) >

Press BACK when you are finished

Figure 24. Lesson -lcexerl-. Options for editing a diagram.

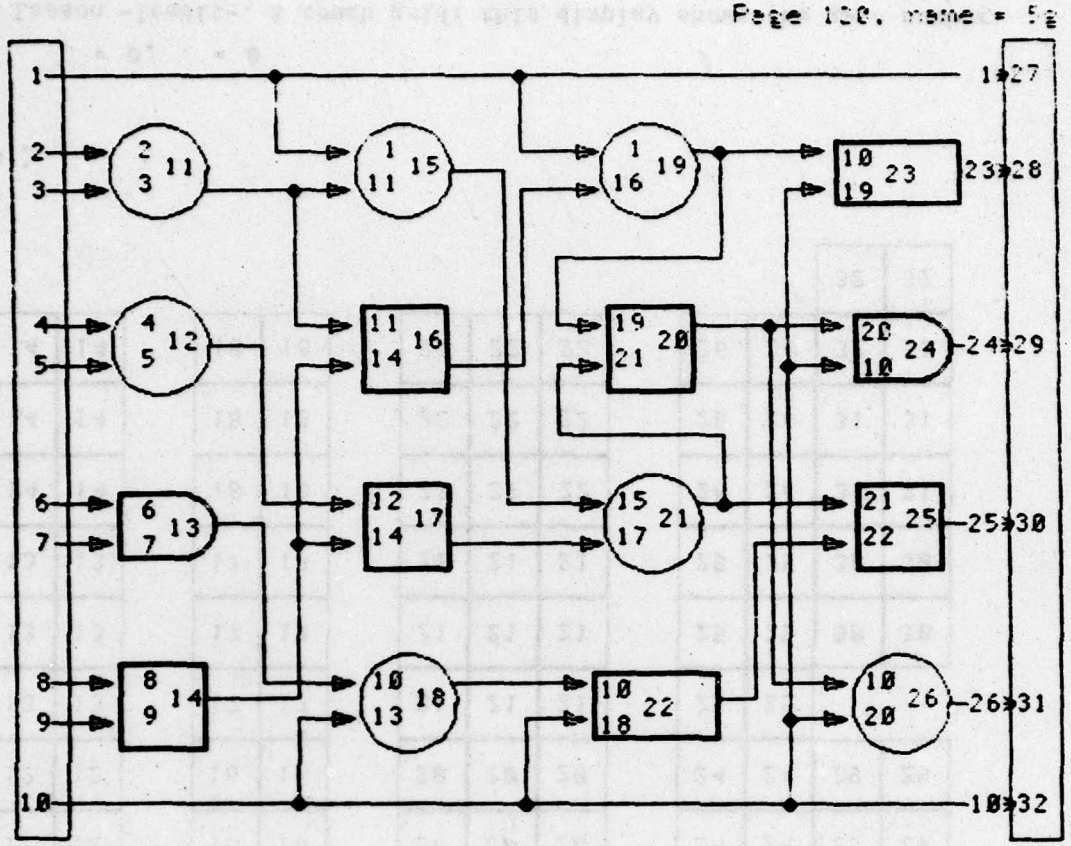


or 11,2,3,56,448,100,423

ENTER 8 >

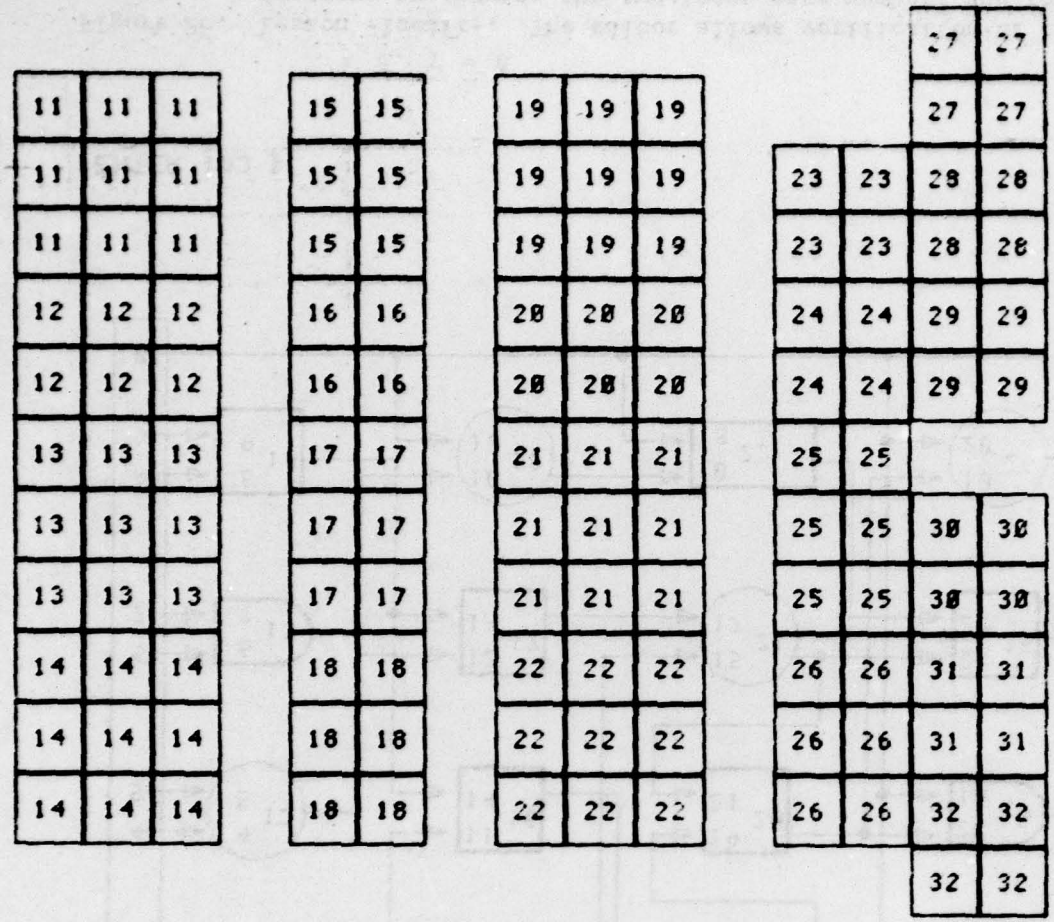
X = 0, Y = 0

Figure 25. Lesson -lcedit-. A circuit diagram displayed by the editor. "Line" 8 is displayed: it is the OR symbol at the upper left of the screen; the cryptic numbers in the "line" stand for, (a) gate number 11; (b) input gates are gates 2 and 3 (both are inputs); (c) the gate is located at 56,448, and the value of the gate ("1") is displayed at 100,423.



X = 0, Y = 0

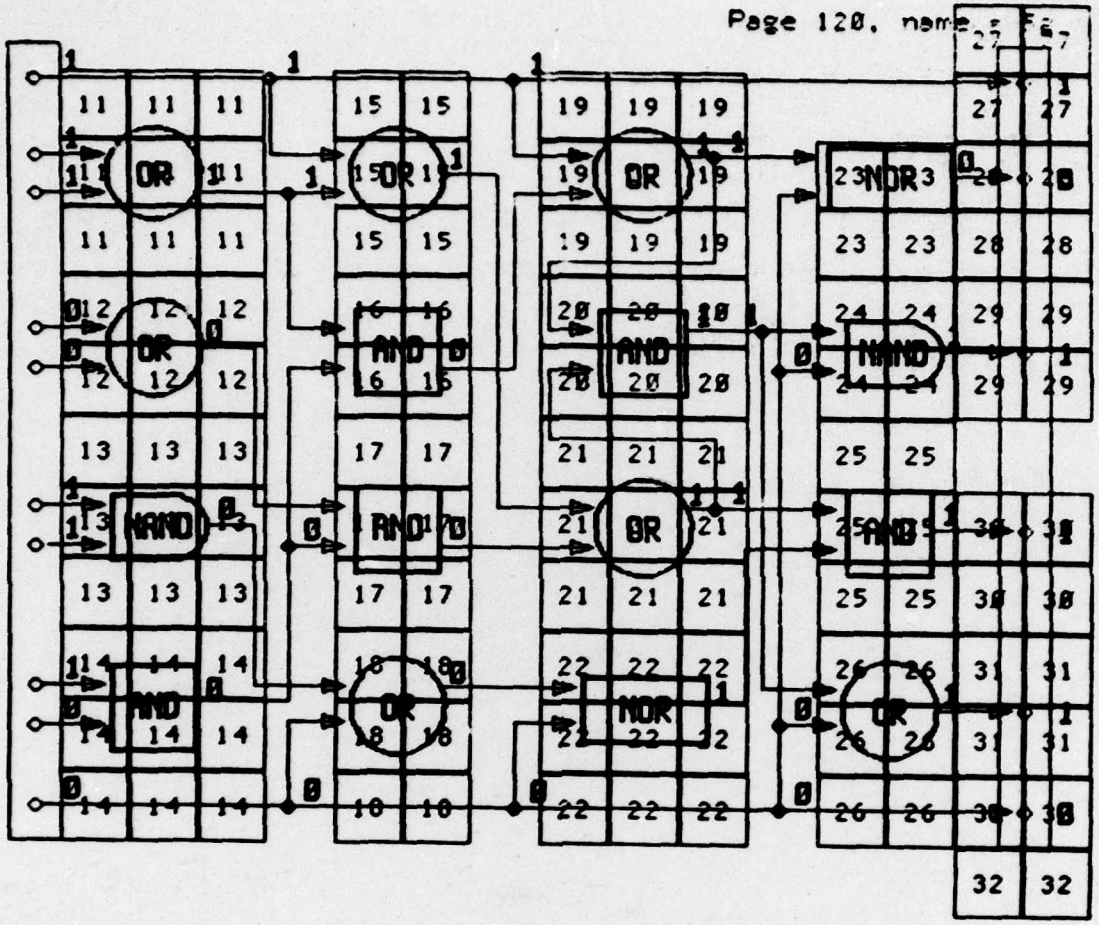
Figure 26. Lesson -lcedit-. The editor allows verification of circuit diagrams by showing the two input gate numbers and the gate number for each gate. Verification is done by visually checking that the input gates are actually attached to the specified gate with line segments.



ENTER 103 >

X = 0, Y = 0

Figure 27. Lesson -lcedit-. A touch grid: this display shows the gate number associated with each screen location. For example, if the students touch anywhere in the upper left hand corner of the screen, it will be assumed that they are touching gate 11.



X = 0, Y = 0

Figure 28. Lesson -lcedit-. A touch grid superimposed on a circuit diagram; this allows verification that each gate has an appropriate touch area around it. The editor allows changing touch areas.